QUINN EMANUEL URQUHART & SULLIVAN, LLP
Sean Pak (Bar No. 219032)
seanpak@quinnemanuel.com
Melissa Baily (Bar No. 237649)
melissabaily@quinnemanuel.com
James Judah (Bar No. 257112)
jamesjudah@quinnemanuel.com
Lindsay Cooper (Bar No. 287125)
lindsaycooper@quinnemanuel.com
Iman Lordgooei (Bar No. 251320)
imanlordgooei@quinnemanuel.com
50 California Street, 22nd Floor
San Francisco, California 94111-4788
Telephone: (415) 875-6600
Facsimile: (415) 875-6700

Marc Kaplan *(pro hac vice)*
marckaplan@quinnemanuel.com
191 N. Wacker Drive, Ste 2700
Chicago, Illinois 60606
Telephone: (312) 705-7400
Facsimile: (312) 705-7401

Attorneys for GOOGLE LLC

UNITED STATES DISTRICT COURT

NORTHERN DISTRICT OF CALIFORNIA

SAN FRANCISCO DIVISION

SONOS, INC.,

Plaintiff,

vs.

GOOGLE LLC,

Defendant.

Case No. 3:20-cv-06754-WHA
Consolidated with Case No. 3:21-cv-07559-WHA

GOOGLE LLC'S RESPONSE TO SONOS'S BRIEF REGARDING THE PARTIES' POSITIONS ON STANDALONE MODE

Google LLC ("Google") provides this response to Sonos's brief regarding Google's expert testimony on the "operating in standalone mode" accused in Google's "new design" products (Dkt. 731). Sonos's objection and motion to strike should be denied for several reasons. First, Sonos made no objections during trial to the introduction of factual testimony and documentary evidence (*e.g.*, source code) from Google's engineers, Ken McKay and Tavis Maclellan, which proved that the "new design" products have ***three*** modes or states of operation: idle mode, standalone mode and group mode. In view of such testimony and documentary evidence regarding the actual design and operation of the "new design" products, Sonos bears the burden of proving infringement on patent claims that require the "zone players" to be in either standalone mode or group mode and exclude a third mode of operation. Second, Google's expert, Dr. Schonfeld provided his non-infringement theory in his expert report based on the same underlying functionality. The only difference was the terminology used by Dr. Schonfeld (*e.g.*, "stops," "terminates") versus the slightly different but equivalent terminology used by Google's engineers (*e.g.*, "idle," "stops" and "kills") to describe the same underlying functionality in the "new design" products. Third, Sonos's own expert Dr. Almeroth understood this exact functionality and provided his opinions based on the existence of three separate states or modes in the "new design" products. Thus, there is no surprise or prejudice to Sonos. At bottom, Google's newly designed products terminate operation of the playback ***application*** (not just pausing playback of music) when they are added to a group.

I. Google's Expert Adequately Described the Changes in the New Design

Sonos's motion concerns ***the words used by*** Google's experts to describe the reasons that the accused products do not meet the "operating in standalone mode" limitation of the asserted claims due to the functionality added by Google to its newly designed speakers. The gist of the key change in the new design is that when a player is added to a *new* speaker group (*existing* speaker groups operate differently), any music ***application*** currently running on the player and every other member of the new speaker group will be stopped (*i.e.*, terminated or killed) and therefore cease to function independently. In other words, a player can be either (1) configured to play music individually, (2) configured to play music in synchrony as a group, or (3) not configured to play music at all, in which case it (obviously) is not playing back individually and is not playing back in

synchrony with a group. The third state is essentially the absence of either of the first two states or modes that have been configured for any type of audio playback. The parties have referred to these three different states in various ways, but as explained below, in the end, the functionality is the same, and Dr. Schonfeld's theory of non-infringement was timely disclosed long before trial.

Central to Sonos's motion is how Dr. Schonfeld described this behavior, but this misses the mark. What matters is the disclosure of the theory and supporting evidence, and not the words used to describe them. Google first provided expert opinions on the StopCurrentApp() function on January 13, 2023, which was shortly after deployment of the new design began. In that report, Dr. Schonfeld opined that in the new design, "[i]n the case where a new group is created, ***the players within that group cease to function as individual players and are instead stopped*** . . . ." Ex. 1 (Schonfeld Reb. Rep.) ¶ 59. Those players then are ***not*** (1) configured to play back music individually or (2) configured to play back music in synchrony as a group, and therefore they are in a third state or mode—which Dr. Schonfeld refers to here as the "cease to function" and "stopped" state. The phrasing does not matter; only the functionality matters. And Dr. Schonfeld accurately described the functionality at issue, using the name of the function in question—***StopCurrentApp()***—to describe the behavior of players added to a group. Dr. Schonfeld repeatedly cited to the ***StopCurrentApp()*** function, which does exactly as its name implies by ***stopping the current app***, not simply pausing or stopping music playback**.** There is nothing hidden or confusing about these opinions.

Of course, the purpose of adding a number of speakers to a speaker group is to actually add those speakers to the group. The next source code function that is called after StopCurrentApp() is therefore "AddGroup()," which performs exactly this role. Dr. Schonfeld opined that "only ***after*** [] playback is ***terminated*** on the speaker is the new group then added through the AddGroup function." *Id.* ¶ 108 (all emphases added unless otherwise noted). In other words, Dr. Schonfeld opined that after the players are "stopped" or "terminated" (per the StopCurrentApp() command), they are then added to a speaker group. Accordingly, Dr. Schonfeld opines in his report that after StopCurrentApp() is called, the speakers "remain stopped in conjunction with the group." *E.g.*, *id.*

¶ 58. A ***stopped*** player that is added to a group is thus ***not*** (1) configured to play back music individually or (2) configured to play back music as a group, and thus it remains in the third state.

II. Sonos Responds to Google's Theories, Making Clear that Sonos Recognized the Various "States" Identified by Dr. Schonfeld

The argument Sonos made in Court is that it has somehow been prejudiced because it was not aware that the StopCurrentApp() function relied on by Google as taking its newly designed products outside the scope of the claims actually ***stopped the current app***, exactly as the function is named. Sonos's counsel argued that prior to trial, "it was entirely about the fact that the, quote, StopCurrentApp function stops ***playback***." *Id.* (emphasis added). Sonos argued that prior to trial it was not about "tearing down" the app or that the app is "brought down." *Id.*

Of course this flies in the face of the actual function name (StopCurrent***App***) and its undisputed functionality according to Google's fact witnesses, but more importantly this position would surprise Sonos's own expert. Dr. Almeorth repeatedly opined prior to trial that the StopCurrentApp() function actually ***stopped the app***, not just that it stopped ***playback*** of music. For example, he opined in his report that:

> if such an Accused Google Player is running a particular receiver app at the time that it receives "join_group" message indicating that the Accused Google Player has been added to a new speaker group (e.g., the YouTube Music receiver app), ***the MultizoneManager::StopCurrentApp() function will cause the Accused Google Player to stop that particular receiver app.***
>
> * * *
>
> the MultizoneManager::StopCurrentApp() function ***causes an Accused Google Player operating in a standalone mode to stop its currently-running receiver app***

Ex. 2 (Almeroth First Suppl. Reply Expert Rep.) ¶ 93 (emphasis added). Dr. Almeroth specifically "confirmed" his "understanding of the MultizoneManager::StopCurrentApp() function" through "the [deposition] testimony of Google's corporate designee, Mr. Kenneth [*sic*] MacKay." *Id.* ¶ 94. Of course, Mr. MacKay has now provided ***exactly the same*** testimony in trial.

Notably, StopCurrent***App*** is not Stop***Playbac***k. Indeed, Sonos asked Mr. MacKay exactly this question in deposition and learned that stopping the app is ***not*** stopping the playback of an app.

> Q. Can you explain for me the practical differences between this StopPlayback() function and the StopCurrentApp() function?
> THE WITNESS: So I don't remember exactly when we call this, but I think -- I think this doesn't – this only stops the playback. So it's as if you're stopping the

> track, the media track from playing, but it doesn't stop the app itself. So the app is still running.
> THE WITNESS: I think StopCurrentApp() is both simpler and covers more cases. ***So it stops not only playback, but it stops the app itself and -- yeah***.

Ex. 3 (MacKay Tr.) at 50:1-20 (objections omitted).

Indeed, Sonos appears to be engaging in a game of semantics. The question is ***not*** how do the parties and their various witnesses refer to the third state or mode of operation in the "new design" products, but rather whether such third state or mode exists in the "new design" products. Sonos's expert refers to the idle state (where the Google speaker is not configured for any playback because the music app has been terminated) in different ways, including an "uninvoked state," but his most straightforward characterization is to call it an "inactive playback ***state***."

> **Almeroth Reply Report, para 128 (Ex. 4)**: Indeed, in such a scenario, an Accused Google Player installed with newly-released firmware version 1.56.324896 behaves in the same way that Accused Google Players installed with prior firmware versions behaved, namely, the Accused Google Player handles the join_group message received from the Accused Google Controller without leaving "standalone mode" or making any change to the ***Accused Google Player's inactive playback state***.

> **Almeroth Reply Report, para 164 (Ex. 4)**: Indeed, in such a scenario, an Accused Google Player installed with newly-released firmware version 1.56.324896 behaves in the same way that Accused Google Players installed with prior firmware versions behaved, namely, the Accused Google Player handles the join_group message received from the Accused Google Controller without leaving "standalone mode" or making any change to the ***Accused Google Player's inactive playback state***.

As noted above, Google's fact witness, Mr. MacKay, refers to this third state or mode as an "idle" mode. And as explained above, Dr. Schonfeld refers to this third state or mode as a "stopped" state and Dr. Almeroth describes the same as a "state of stopped playback." In the end, there are actually no *functional* differences between any of the experts' or fact witnesses' understanding on this point. For clarity, Google summarizes the different characterizations of the third playback mode in a table below.

Mr. MacKay	Dr. Schonfeld	Dr. Almeroth
Idle state	Stopped	Inactive playback state
Idle mode	Terminated	Uninvoked state[1]
Inactive	State of stopped playback	Unlaunched state

III. Sonos's Arguments Lack Merit

First, Sonos argues that Dr. Schonfeld only disclosed theories that a standalone mode speaker would stop playback and not stop its application. As shown above, this is incorrect because Dr. Schonfeld did describe a player as being stopped and existing in a different "state" than either individual playback mode or grouped playback mode. The fact that Dr. Schonfeld referred repeatedly to the fact that speakers that have their music stopped is nothing but an obvious consequence of stopping or "terminating" the ***application***. If there is no application running, there is no playback from that (non-existent) application.

Next, Sonos argues it would be prejudiced by Google's "late" disclosure of its position for the first time at trial,[2] but even assuming *arguendo* that Sonos could show that Google's experts did not adequately disclose his non-infringement theory based on a third mode or state (which is not the case), Google's position was no secret. Indeed, Google could not have articulated this any more clearly than in its summary judgment briefing in this case:

> Sonos's opposition misses the key reason why Google's redesigned speakers cannot and do not infringe: ***Google's speakers are not always in "standalone mode" or "group mode." Rather, they can be, and often are, inactive (not in either mode) because a user has not yet chosen a playback mode.*** Ex. 7 ¶¶ 3-4.

Dkt. 536-3 at 11 (cleaned up).

[1] Almeroth Rep. ¶ 111 ("the new speaker group did not previously exist and is in an uninvoked state at the time of creation"); *id.* ¶ 109 ("where the pre-existing speaker group would remain in an *uninvoked state* after the Accused Google Player is added"); *id.* ¶113 ("the new speaker group starts out in an *uninvoked state* (or an unlaunched state in Google's terms).

[2] Sonos also argues that Google should have provided more disclosure in its interrogatory responses, Br. at 2, but Sonos overlooks the fact that discovery closed on November 30, 2022. Had Google supplemented its interrogatory responses, Sonos would simply have objected to those disclosures as new positions and moved to strike.

Dated: May 16, 2023

Respectfully submitted,

/s/ Sean Pak
Attorneys for GOOGLE LLC

QUINN EMANUEL URQUHART & SULLIVAN, LLP

Counsel for Google LLC

ATTESTATION

Pursuant to the Federal Rules of Civil Procedure and Local Rule 5-1, I hereby certify that, on May 16, 2023, all counsel of record who have appeared in this case are being served with a copy of the foregoing via the Court's CM/ECF system and email.

DATED: May 16, 2023

By: */s/ Sean Pak*
Sean Pak